Internet Content Adaptation Protocol (ICAP)

Network Appliance

Version 1.01

7/30/01

Contents

1. Scope/Executive Summary

This document provides a brief introduction to ICAP, explains what ICAP means to the future of Internet content handling, and discusses how NetCache is ideally suited to handle these ICAP enabled value-added services. The following section introduces ICAP, shows why it is needed, and how it can be leveraged to standardize and modularize Internet content manipulation (content vectoring).

2. Introduction

ICAP was formally introduced in December 1999 by the ICAP Forum (www.i-cap.org). The ICAP Forum is a coalition of Internet businesses covering a wide array of services; including hardware and software providers, content distribution providers, application service providers, advertising institutions, hosting providers, broadband providers, etc. These members are dedicated to making the future of the Internet accessible (and scalable) to the evolving value-added services (services added onto plain old Internet access) business. This forum was co-founded (and still co-chaired) by Network Appliance and Akamai Technologies. What are the

goals of the ICAP Forum?

- Make content more flexible for end users.
- Provide a common, open standard for edge-based appliance communication to handle these value-added services.
- Off-load resource-intensive APIs and other services from Web site servers to dedicated servers.

The protocol needs to:

- Be simple.
- Be scalable.
- Use existing infrastructure.
- Be modular in its service. That is, services must be able to be added and subtracted without affecting the intervening architecture or its performance.
- Use existing communication methods and standards.
- Provide resource savings by leveraging edge services.

ICAP is a protocol designed to off-load specific Internet-based content to dedicated servers, thereby freeing up resources and standardizing the way in which features are implemented. For example, a server that handles only language translation is inherently more efficient than any standard Web server performing many additional tasks. ICAP concentrates on leveraging edge-based devices (proxies and caches) to help deliver value-added services. At the core of this process is a cache that will proxy all client transactions and will process them through ICAP/Web servers. These ICAP servers are focused on a specific function, for example, ad insertion, virus scanning, content translation, language translation, or content filtering. Off-loading value-added services from Web servers to ICAP servers allows those same web servers to be scaled according to raw HTTP throughput versus having to handle these extra tasks.

ICAP in its most basic form is a "lightweight" HTTP based remote procedure call protocol. In other words, ICAP allows its clients to pass HTTP based (HTML) messages (Content) to ICAP servers for adaptation. Adaptation refers to performing the particular value added service (content manipulation) for the associated client request/response.

2.1 What Initiated ICAP?

The problem with the Internet, in its early stage of growth, is the ability to scale the traffic, bandwidth, services, access, etc. If it’s possible to do something over the Internet, odds are it won’t scale to meet the demand. This certainly is the case for value-added services. These services provide real and necessary applications for clients. Unfortunately, these value-added services consume resources such as bandwidth and server processing time, and therefore

introduce more delay into the already "click and wait" process.

Today's services almost exclusively run on proprietary APIs that are custom built for the provider's particular business application. These APIs are often unreliable because they aren't designed to scale with the hardware as the "e-commerce" companies' Web business grows. In addition, these APIs can be very costly to change and any new service would require an additional API.

Over-Taxed Servers. Today's laundry list of services that are offered can overload and slow down a site's access and transactions to a point of losing business. These value-added services each tend to run through a separate software application, causing servers to bog down. The servers are not appliances designed from the ground up to run these APIs, but they are asked to handle these services on top of what they already are asked to do. Such services include access, authentication, customer information database, e-commerce, language translation, content filtering virus scanning, ad insertion, etc., all leading to higher latency and reduced reliability.

Too Much Overhead. In addition to bogging down servers, the value-added services cause additional latency risks by clogging bandwidth and the network highways that must carry all this traffic.

Wireless Device Support. Supporting many of the same Web services for wireless customers that are available for the PC user, such as synchronizing e-mail between all your latest devices, has become a necessary service. With the proliferation of wireless devices (expansion of the wireless market), there exists too many communication varieties (standards) that must be adapted for communication into and out of the Internet. Adding to this problem is the relatively narrow transmission rate (9.6kbps on average). The solution has been a collection of network devices that include transmission towers, wireless gateways, proxy servers, and additional applications running on top of origin servers (content) that are adapted to scale down their content and services for the small amount of data these devices can handle.

2.2 Solution

What is the answer to the above problems? ICAP solves the above problems with its open architecture and ease of the modular development process.

2.2.1 Benefits of ICAP

- ICAP leverages existing equipment available today. In fact, if NetCache (a proxy appliance) proxies are already installed, then no new equipment is necessary, with the exception of the ICAP servers.

- ICAP is HTTP based, enabling access through security barriers that only allow port 80 traffic. Therefore, no security changes to the existing network are likely.

- ICAP is an *open protocol* and allows any server or application provider to implement it. ICAP is easy to implement since it leverages Apache code. ISPs and enterprises can then choose the appropriate value-added application provider.

- ICAP can also collect client interest information for use in targeting more focused advertising toward these individuals.
- ICAP off-loads these value-added services to ICAP servers, freeing up the resources of the Web servers. This reduces the access times on these sites.
- ICAP simplifies the implementation, reliability, and scalability of value-added services. ICAP leverages edge device and infrastructure to deliver edge-based value-added services that require content adaptation.

2.2.2 What Do All These Benefits Give You?

- Able to implement services quickly
- Able to outsource a service completely
- Improve user satisfaction
- Target ads better and cheaper
- Improve management, security, and control of content
- Optimize site scalability, reliability, and performance
- Derive new revenue streams

2.3 Example ICAP Services

2.3.1 Virus Scanning

This is the ability to perform "on-the-fly" virus checks of new content and provide cached content that was previously scanned.

Historical Method: Virus scanning was always left to the receiving network (or PC) to accomplish, and every object has the potential to be scanned many times, causing a waste of resources. There is no historical "on-the-fly" method for virus scanning prior to delivery.

What are the benefits under ICAP? ICAP's "on-the-fly" virus scanning allows previously scanned (and unaffected) objects to be cached and provided virus free.

2.3.2 Markup Language Translation (PDAs/Cell Phones).

This is the ability to allow non-HTML devices (such as cellular phones) to talk to HTML devices such as PCs and vice versa. This is essentially a WAP/XML/HTML translation bridge.

Historical Method: Deploy gateways specific to translating the particular device's language to HTML and back. This method involves funneling all transmissions into a single set of gateways as a point of presence to the Internet.

What are the benefits under ICAP? Under ICAP, point-of-presence entry can reside anywhere a transmission point can tap into the network. A cache can handle all client requests through redirects to translation ICAP servers and maintain cached copies of multiple formatted objects for faster response to the client.

2.3.3 Advertising Insertion

This is the ability to insert ads into Web pages or to spawn new pages based on customer preferences/history/location when a customer performs a request from a Web site such as a search engine.

Historical Method: The historical method for ad insertion was based on either the origin Web site's ISP, the hosting provider, or the site itself signing up for direct advertising.

What are the benefits under ICAP? Ad insertion will be more focused to individuals based on originating IP address of the proxy server, customer-entered keywords (for example, typing in *Star Wars* may get the customer a direct ad from Amazon or a local bookshop), or customer collected information (profiling). Profiling can be very specific. For example, an individual browsing *Scientific American* may click on a car ad and later click on a car ad in *National Geographic*. Based on what type of car ad was queried by the client, a specific ad could be targeted at the client for a local car dealership that has that car for sale and it's current price.

- Geography (IP address/zip code)
- Search engine
- Keyword adaptation
- Customer profiling

2.3.4 Human Language Translation

This is the ability to translate formatted HTML-tagged content from one language to another (for example, English to Japanese).

Historical Method: Expansive and resource-hungry APIs running typically on client machines or Web servers. Most of these services are manual and limited.

What are the benefits under ICAP? Under ICAP, certain originating requests will by geography or direct input be translated by ICAP servers via redirection from proxy servers.

2.3.5 Content Filtering

This is the ability to redirect an unauthorized or restricted request to another site/page.

Historical Method: Performed by the proxy server via manually entered information or downloaded (subscribed database) from a site-list filtering reseller (for example, smart filter).

What are the benefits under ICAP? Under ICAP, the content filtering is more extensible from a

customer perspective. Now dynamic content can be filtered and both content filtering and the management of the service can be out-sourced. In addition, the filtering (ICAP) servers can be located remotely from a customer's site.

2.3.6 Data Compression

This is the ability to Compress HTML pages or objects from an origin server.

Historical Method: No HTML compression and compression accomplished manually for embedded objects.

What are the benefits under ICAP? Under ICAP, origin server responses can be compressed allowing bandwidth to be saved.

2.4 ICAP Policies

Generally, ICAP does not specify the when, who, or why for content manipulation, but only how to make content available to an application server that will perform the adaptation. For example, if ICAP is the tool to allow content translation/adaptation, you will still need an adaptation engine (ICAP server) to decide when, who, or why.

3. ICAP Architecture

How is ICAP architected? Since many services are expected to leverage this protocol, a modular, simple, and "easy to implement" schema is needed. This protocol must also communicate using existing methods and therefore be completely compatible with the installed base of standard network devices and existing standard applications.

Commercial Web sites are sets of Web, e-commerce, file, and FTP servers and databases. Although communication is not restricted to just HTTP, the vast majority of layer 7 traffic is HTTP based. ICAP proxies simply use an HTTP post in which "client request" and "propose origin server response" are encapsulated within the first part of the HTML body.

Through the use of the following four adaptation techniques, ICAP is able to facilitate all the necessary content adaptation. In each case a cache (forward proxy) acts as the central point in ICAP rout processing (page parsing) and initial capture and final response to the client. The cache is able to cache portions of the ICAP for future provision without the need to secure a new copy or reprocess the request.

3.1 Request Modification

Summary: The client's request is redirected to an ICAP server that modifies the request prior to being fulfilled by the origin server.

Details: In this model, a client sends a request to an origin server. This request is redirected to an ICAP server by the intervening proxy server (cache). The ICAP server modifies the message and sends it back to the proxy server. The proxy server parses the modified message and forwards it

to the origin server to fulfill the client's request. The request is then executed by the origin server and the response delivered to the client.

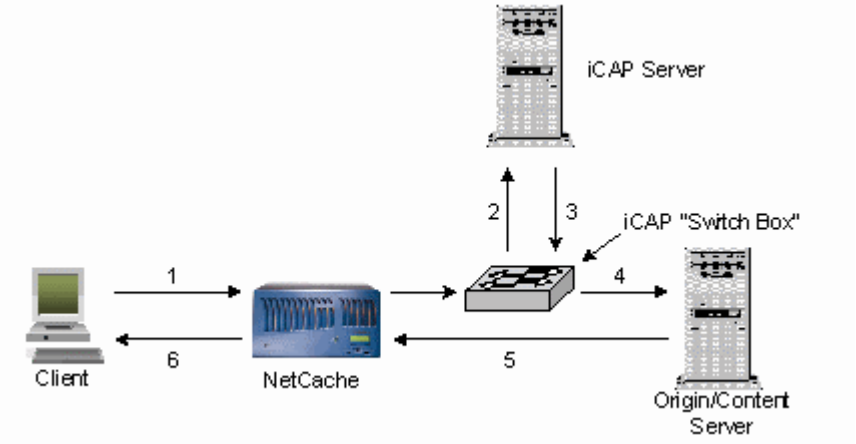

Figure 1: Request Modification

Example: Content Filtering. The client sends out a request for a Web page and the proxy server redirects that request to the ICAP server. The ICAP server parses the HTML request and performs URL-based filtering by comparing the request URL to a list of "banned" URLs. If the URL is on the "banned" list, then the client's request is modified to request an error message from the origin server or, more likely, from the proxy server (cache). This error message is then supplied to the client. If the origin server URL was not banned, the ICAP server would forward the request to the origin server via the proxy server and the request would be fulfilled.

3.2 Request Satisfaction

Summary: The client's request is redirected to an ICAP server that modifies the request prior to being fulfilled by the origin server. The modified request is sent directly to the origin server without returning it to the proxy server first (as is done in Request Modification).

Details: In this mode, a client sends a request to an origin server. This request is redirected to an ICAP server by the intervening proxy server (cache). The ICAP server modifies the message and sends it straight to the origin server for fulfillment. After processing the request, the origin server sends it back to the client via the ICAP server and proxy server.

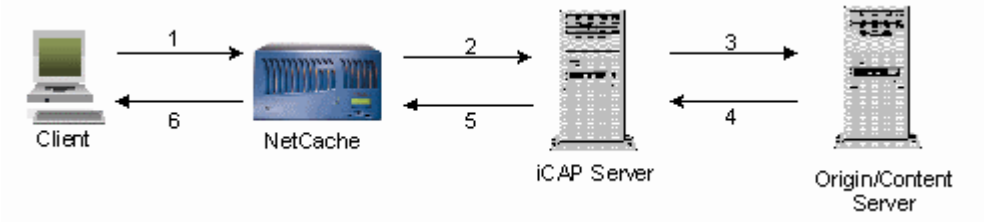

Figure 2: Request Satisfaction

Example: Using the same content filtering example above, the process changes with request modification. The client's request is still examined by the ICAP server, but if the content is not authorized, the ICAP server will send back an error response through the proxy server and ignore the origin server. If the client is authorized to access the origin server, then the ICAP server will fetch the objects from the origin server and provide them to the client.

3.3 Response Modification

Summary: The client's request is processed by the origin server, but the ensuing response is redirected to the ICAP server for modification prior to delivery to the client.

Details: In this mode, a client sends a request to an origin server. The request is fulfilled as would be expected by the origin server. The response, however, is redirected by the proxy server to the ICAP server. The ICAP server modifies the response message and delivers it to the client via the proxy server.

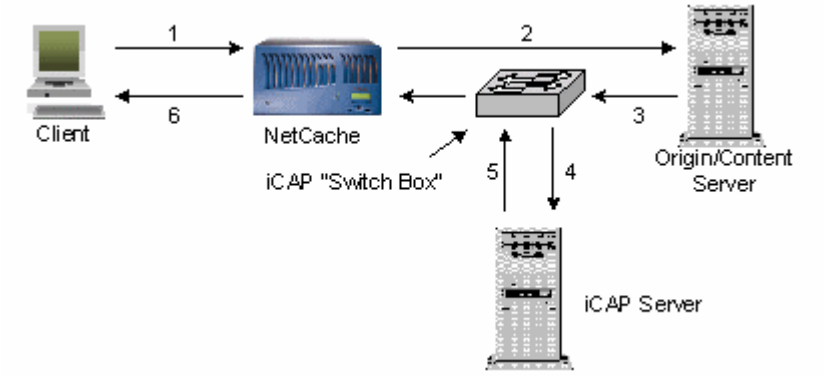

Figure 3: Response Modification

Example: Gateway Translation (HTML Formatting). A request is made by cellular phone for a company's stock profile. The request is forwarded to the origin server, which, fulfills the request. The response by the origin server, however, is redirected to an ICAP server which modifies the response to allow the cell phone to display the response properly. Note that the incoming request may have to be modified to begin with (request modification).

3.4 Result Modification

Summary: The client's request is processed by the origin server, and the ensuing response is redirected to the ICAP server for modification. This differs from Response Modification in that the ICAP server is downstream of the proxy cache.

Details: In this mode, a client sends a request to an origin server. The request is fulfilled as would be expected by the origin server. The response, however, is redirected downstream of the proxy server to the ICAP server. The ICAP server modifies the response message and delivers it to the client via the proxy server. The advantage of this method over response modification is that since the ICAP server is down-stream of the proxy, the proxy can cache objects that would be delivered to the client.

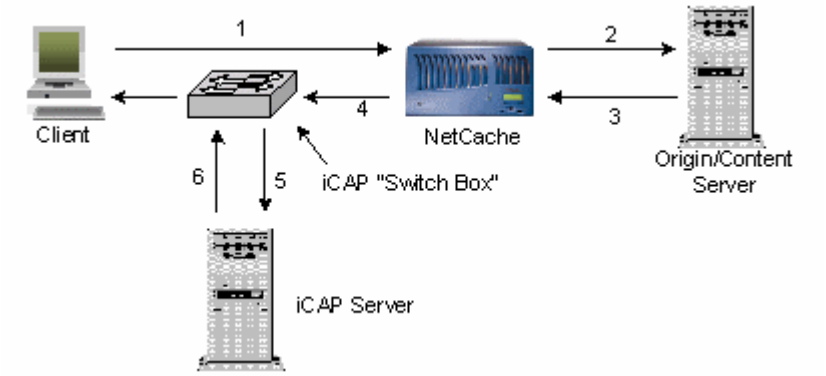

Figure 4: Result Modification

Example: Ad insertion. The client sends out a request for a Web page and the proxy server directs that request to the origin server. The origin server fulfills the request and delivers objects to the proxy. The proxy caches the objects and directs the response to an ICAP ad server. The ICAP server parses the HTML response and performs profiling of the client. The ICAP server inserts appropriate targeted adds and sends response to client.

The following table summarizes the services available through each architecture form.

Service	Architecture			
	Request Modification	Request Satisfaction	Response Modification	Result Modification
Content Filtering	Yes	Yes	Yes	Yes
Gateway Translation	Yes		Yes	
Language Translation	Yes	Yes	Yes	
Virus Scanning			Yes	
Ad Insertion	Yes	Yes	Yes	Yes
Data Compression			Yes	Yes

Table 1: Service Architecture Summary

3.5 Protocol Details

3.5.1 Communication

ICAP communicates via TCP sessions. The ICAP server is passively listening for any redirected requests as would be expected from any normal Web server. All ICAP message formats conform to RFC 822.

3.5.2 Vectoring Points

The three architecture descriptions mentioned above cover the initial applications that will be deployed for ICAP, but the actual ICAP process can be thought of as vectoring points (potential modification nodes). A node will be referred to as a network device that processes HTML messages. For future implementations of ICAP, these vectoring points will be the basis for the appropriate ICAP-developed process.

- Modification of requests coming into a proxy server (cache)
- Modification of responses coming into to a proxy server (cache)
- Modification of requests coming into an ICAP server
- Modification of responses coming into an ICAP server

4. NetCache Deployment in Support of ICAP

NetCache is a proxy appliance that reduces bandwidth load and latency. Such a device is necessary as the center device in any ICAP implementation. The deployment of NetCache as an ICAP gateway enables ICAP to function. In addition, NetCache is a content *caching appliance* that is scalable, supports all communications protocols, and is transparent to the end user if used with a layer 4 or 7 switch or a WCCP-enabled router.

Using NetCache for ICAP:

- NetCache is an appliance built from the ground up to serve data, not an application running on a separate OS. This makes NetCache far more reliable and much faster in performance than software-based proxies.
- NetCache is an edge-based device close to the user. This reduces latency time and saves bandwidth.
- NetCache caches ICAP information and HTML objects, and can also cache virtually anything that is a file. Adaptations of content can be performed near the edge of a network instead of requiring an updated copy of an object from an origin server or even

an ICAP server. Previous manipulation of objects need not be repeated, as NetCache can provide the objects quickly at the edge.

- What impact does ICAP have on the performance of the cache? Since NetCache is just a proxy/cache, there is very little performance impact.

4.1 Why Choose NetCache?

NetCache provides the following functionality and value-added resources in addition to ICAP:

- NetCache is far more reliable than a Windows NT® or UNIX® server – 99.99+% up-time.
- NetCache is an appliance designed from the ground up to be a cache and not a software package running on top of another operating system.
- NetCache eliminates stale content across geographically distributed caches.
- NetCache can handle thousands of simultaneous connections.
- NetCache can handle hundreds of megabits/sec of data throughput.
- NetCache offers outstanding proxy-based security.
- NetCache can control content of Web servers through its Eject/Pre-fetch feature.
- NetCache works seamlessly with firewalls and can distribute traffic over multiple firewalls.
- NetCache deploys and works seamlessly with other network components.
- NetCache frees up resources of the Web server it's accelerating.
- NetCache supports LDAP/Radius Authentication.
- NetCache can filter out non-work-related Internet content.
- NetCache can split live streaming media to thousands of clients using all major protocols.
- NetCache can cache video/audio on-demand streams for all major media protocols

5. Conclusion

NetCache provides a high-performance, ICAP-enabled cache proxy appliance. NetCache is 99.99+% reliable, easy to install, highly scalable, and ideal for distributing load at the edge of the network. NetCache handles many more client connections and has a much higher data throughput than conventional network devices. Therefore, the NetCache line of products is fully capable of handling all ICAP and caching needs of application service providers, data centers,

content providers, and hosting companies.

Glossary

HTTP

Hypertext Transport Protocol [RFC-1945,RFC-2616], the protocol most often used to transport Web pages.

IP

Internet Protocol, the lowest-level protocol used across all portions of the Internet.

ISP

Internet Service Provider.

L4 switch

A network device that switches packets based on information from layer 4 of the ISO 7-layer model.

LAN

Local Area Network, a high-speed network used within a building or campus. Compare to WAN.

POP

Point-of-Presence, a location where a customer may connect to an ISP.

Response

An HTTP response message.

Request

An HTTP request message.

TCP

Transmission Control Protocol, a connection-oriented protocol layered on top of IP.

URL

Uniform Resource Locator [RFC-1738], a concise notation used to specify the location of a resource or object on the Internet.

WAN

Wide-Area Network, a network connecting multiple LANs, often over a large geographic area.

WCCP

Web Cache Control Protocol.

References

[RFC-1738]

T. Berners-Lee, L. Masinter, M. McCahill, *Uniform Resource Locators (URL)*, December 1994.

[RFC-1945]

T. Berners-Lee, R. Fielding, H. Frystyk *Hypertext Transfer Protocol -- HTTP/1.0*, May 1996.

[RFC-2616]

R. Fielding et al., *Hypertext Transfer Protocol -- HTTP/1.1*, June 1999.

RFC-draft

ICAP Forum, ICAP Draft.